

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-108662

(43)Date of publication of application : 12.04.2002

(51)Int.Cl.

G06F 12/00

G06F 3/06

G06F 17/30

(21)Application number : 2000-301676

(71)Applicant : FUJITSU LTD

(22)Date of filing : 02.10.2000

(72)Inventor : YAMANAKA MINORU

AJISAWA SUSUMU

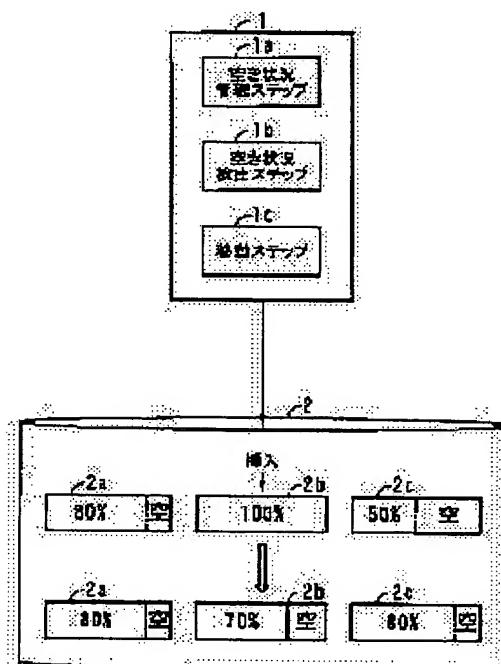
SEKIYA NOBUYUKI

(54) METHOD FOR MANAGING INFORMATION

(57)Abstract:

PROBLEM TO BE SOLVED: To execute a system for defragging information constituted by continuously arranging pages constituted of plural data according to a fixed logical relation without stopping this system.

SOLUTION: An idle situation management step 1a manages the idle situation of pages 2a-2c stored in a data base 2. An idle situation detection step 1b detects the idle situations of the pages 2a and 2c before and after the target page 2b at the time of adding data to the page by referring to the idle situation management step 1b. A movement step 1c moves a part of data included in the page 2b to the page before or after the page 1b according to the idle situations of the pages before and after the page 2b detected by the idle situation detection step 1b, and ensures the space area. Then, new data are inserted into the ensured space area.



LEGAL STATUS

[Date of request for examination]

27.08.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-108662

(P2002-108662A)

(43) 公開日 平成14年4月12日 (2002.4.12)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 F 12/00	5 0 1	G 0 6 F 12/00	5 0 1 S 5 B 0 6 5
	5 2 0		5 2 0 A 5 B 0 7 5
3/06	3 0 2	3/06	3 0 2 J 5 B 0 8 2
17/30	2 4 0	17/30	2 4 0 A

審査請求 未請求 請求項の数 5 O L (全 10 頁)

(21) 出願番号 特願2000-301676(P2000-301676)

(22) 出願日 平成12年10月2日 (2000.10.2)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 山中 稔

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 味沢 丞

神奈川県小田原市栄町一丁目6番地4 株
式会社富士通ソフトウェア生産技術研究所
内

(74) 代理人 100092152

弁理士 服部 毅蔵

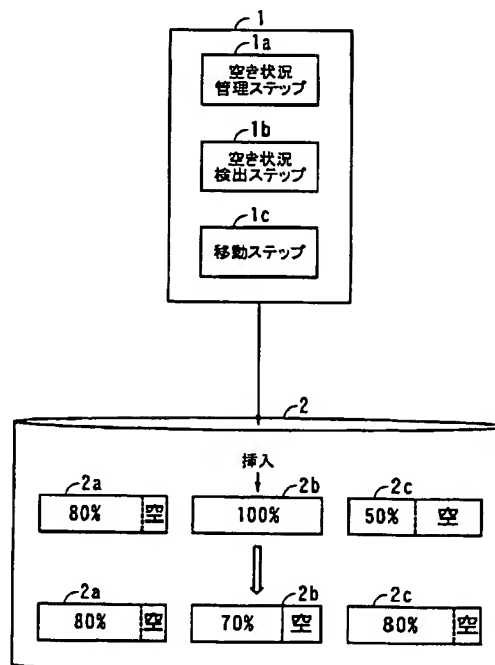
最終頁に続く

(54) 【発明の名称】 情報管理方法

(57) 【要約】

【課題】 複数のデータから構成されるページが一定の論理関係に従って連続的に配置されてなる情報のデフラグをシステムを停止することなく実行する。

【解決手段】 空き状況管理ステップ1 aは、データベース2に格納されているページ2 a~2 cの空き状況を管理する。空き状況検出ステップ1 bは、ページにデータを追加する場合には、対象となるページ2 bの前後のページ2 a, 2 cの空き状況を、空き状況管理ステップ1 bを参照して検出する。移動ステップ1 cは、空き状況検出ステップ1 bによって検出された前後のページの空き状況に応じて、ページ2 bに含まれているデータの一部を前または後のページに移動し、空き領域を確保する。そして、確保された空き領域に対して新たなデータを挿入する。



【特許請求の範囲】

【請求項 1】 複数のデータから構成されるページが一定の論理関係に従って連続的に配置されてなる情報を管理する情報管理方法において、

各ページの空き状況を管理する空き状況管理ステップと、

何れかのページにデータを追加または削除する場合には、対象となるページの前後のページの空き状況を、前記空き状況管理ステップで得られた各ページの空き状況を参照して検出する空き状況検出ステップと、
前記空き状況検出ステップによって検出された前後のページの空き状況に応じて、データの挿入前またはデータの削除後のページに含まれているデータを、前または後のページに移動させる移動ステップと、
を有することを特徴とする情報管理方法。

【請求項 2】 前記移動ステップは、新たなデータを挿入する場合には、前記空き状況検出ステップによって検出された前後のページの空き状況に応じて、前記挿入対象となるページに属するデータを、前または後のページに移動させることを特徴とする請求項 1 記載の情報管理方法。

【請求項 3】 前記移動ステップは、データを削除する場合には、前記空き状況検出ステップによって検出された前後のページの空き状況に応じて、削除の対象とならなかったデータを前または後のページに移動させることを特徴とする請求項 1 記載の情報管理方法。

【請求項 4】 前記空き状況管理ステップは、空き領域が所定の容量以下である場合には、ページに空き領域が存在しないとみなすことを特徴とする請求項 1 記載の情報管理方法。

【請求項 5】 複数のデータから構成されるページが一定の論理関係に従って連続的に配置されてなる情報をコンピュータに管理させるプログラムが記録されたコンピュータ読み取り可能な記録媒体において、

コンピュータを、
各ページの空き状況を管理する空き状況管理手段、
何れかのページにデータを追加または削除する場合には、対象となるページの前後のページの空き状況を、前記空き状況管理手段を参照して検出する空き状況検出手段、

前記空き状況検出手段によって検出された前後のページの空き状況に応じて、データの挿入前またはデータの削除後のページに含まれているデータを、前または後のページに移動させる移動手段、
として機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は情報管理方法に関し、特に、複数のデータから構成されるページが一定の

論理関係に従って連続的に配置されてなる情報を管理する情報管理方法に関する。

【0002】

【従来の技術】 データベースシステム等のデータの書き込みまたは削除を伴うシステムでは、使用時間の経過とともに、2 次記憶装置におけるデータの配置が断片化し、アクセス性能の低下を招く結果となることが知られている。

【0003】 このような断片化が発生する一例として、データベースの索引用の B-T r e e (Balance Tree) について以下に説明する。図 10 は、B-T r e e の一例を示す図である。この図に示すように、B-T r e e は、ルートページ、インデックスページ、および、リーフページから構成されており、ルート（根）からリーフ（葉）に至るに従って枝分かれする木構造を有している。各枝を結ぶ節の部分にはページが配置されており、ルートページおよびインデックスページには、キー値に基づいて該当するリーフに到達するための情報として、範囲を示す情報が格納されている。また、リーフページには、該当するレコードを特定するためのインデックスが格納されている。

【0004】 具体例として、キー値が“10”の場合について考えると、まず、ルートページでは、0～50 の範囲が“50”によって示され、また、51～100 の範囲が“100”によって表されているので、“10”は左側の“50”の範囲に該当することになり、左側のインデックスページに進む。

【0005】 左側のインデックスページでは、前述の場合と同様の判断により、“30”の範囲に該当するので、最左端のリーフページに進むことになる。最左端のリーフページでは、インデックスは“10”と“30”であり、いまの例では、“10”に該当するので、このインデックスに関連付けられたレコード（2 次記憶装置の記憶位置）に至ることができる。

【0006】 ところで、以上のような B-T r e e によって管理されたデータベースに対して新たにデータを追加したり、既存のデータを削除する場合には、B-T r e e 自身に対しても変更を加える必要が生じる。そのような場合の具体例について以下に説明する。

【0007】 図 11 は、新たなデータをデータベースに追加する場合において、リーフページに対して行う一連の手続きを示す図である。図 11 (A) に示すように、連続するリーフページ 10～12 が存在する場合に、リーフページ 11 に対して値が“60”であるインデックス 20 を追加する場合について考える。この場合、追加の対象となるリーフページには、空きが存在しないので、まず、新たなリーフページ 13 が確保される。そして、図 11 (B) に示すように、この新たなリーフページに対して、リーフページ 11 の 2 つのインデックスが移動される。

【0008】そして、図11(C)に示すように、空きが発生したリーフページ11に対して、新たなインデックス20が書き込まれることになる。図12は、既存のデータをデータベースから削除する場合において、リーフページに対して行う一連の手続きを示す図である。

【0009】図12(A)に示すように、連続するリーフページ10~12が存在する場合に、リーフページ11から、値が“50”であるインデックスを削除する場合について考える。この場合、このインデックスを削除すると、リーフページ11にはインデックスは存在しないことになるので、図12(B)に示すように、このページは縮退ページになり、システムに対して返還されることになる。

【0010】このように、B-Treeでは、新たなインデックスが追加された場合には、図11に示すように、空の部分を持つリーフページ(断片化)が発生する場合があります。このような空の部分を持つリーフページは、新たなインデックスが格納されて一杯になるか、または、図12に示すように、インデックスが削除されて縮退ページとしてシステムに返還されるまで存在することになっていた。なお、このような断片化の問題は、インデックスページについても発生していた。

【0011】従来においては、このような断片化が発生した場合には、システムの運用を一旦停止し、デフラグ用のアプリケーションプログラムを実行することにより、データの再配置を行っていた。

【0012】

【発明が解決しようとする課題】しかし、フラグメンテーションは、運用状況によって発生頻度が変化するので、システムの管理者が発生状況を監視する必要があるという問題点があった。

【0013】また、近年では、データベース等は24時間無停止で運用される場合が多いため、デフラグのためにシステムの運用を停止することは困難であるという問題点もあった。

【0014】本発明はこのような点に鑑みてなされたものであり、システムの管理者の負担を軽減するとともに、24時間運用がなされているシステムでも使用可能な情報管理方法を提供することを目的とする。

【0015】

【課題を解決するための手段】本発明では上記課題を解決するために、図1に示す、複数のデータから構成されるページ2a~2cが一定の論理関係に従って連続的に配置されてなる情報を管理する情報管理方法において、各ページの空き状況を管理する空き状況管理ステップ1aと、何れかのページにデータを追加または削除する場合には、対象となるページの前後のページの空き状況を、前記空き状況管理ステップ1aで得られた各ページの空き状況を参照して検出する空き状況検出ステップ1bと、前記空き状況検出ステップ1bによって検出され

た前後のページの空き状況に応じて、データの挿入前またはデータの削除後のページに含まれているデータを、前または後のページに移動させる移動ステップ1cと、を有することを特徴とする情報管理方法が提供される。

【0016】ここで、空き状況管理ステップ1aは、各ページの空き状況を管理する。空き状況検出ステップ1bは、何れかのページにデータを追加または削除する場合には、対象となるページの前後のページの空き状況を、空き状況管理ステップ1aで得られた各ページの空き状況を参照して検出する。移動ステップ1cは、空き状況検出ステップ1bによって検出された前後のページの空き状況に応じて、データの挿入前またはデータの削除後のページに含まれているデータを、前または後のページに移動させる。

【0017】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照して説明する。図1は、本発明の動作原理を説明する原理図である。この図に示すように、本発明の情報管理方法に係るデータ管理装置1は、空き状況管理ステップ1a、空き状況検出ステップ1b、および、移動ステップ1cによって構成されており、データベース2に接続されている。

【0018】ここで、空き状況管理ステップ1aは、各ページの空き状況を管理する。ここで、ページとは、一定量のデータを格納することが可能な記憶領域をいう。空き状況検出ステップ1bは、何れかのページにデータを追加または削除する場合には、対象となるページの前後のページの空き状況を、空き状況管理ステップ1aで得られた各ページの空き状況を参照して検出する。

【0019】移動ステップ1cは、空き状況検出ステップ1bによって検出された前後のページの空き状況に応じて、データの挿入前またはデータの削除後のページに含まれているデータを、前または後のページに移動させる。

【0020】データベース2には、複数のデータから構成されるページ2a~2cが一定の論理関係に従って連続的に配置されてなる情報が格納されている。次に、以上の原理図の動作について説明する。

【0021】以下では、データベース2に格納されているページ2a~2cに注目し、ページ2bに対して新たなデータを追加する場合について考える。空き状況検出ステップ1bは、挿入対象となるページ2bの空き状況を、空き状況管理ステップ1aを参照して検出する。この例では、データベース2内部の上側に示すように、対象となるページ2bは100%データが格納されているので、空きが存在しないことが検出される。

【0022】対象となるページ2bに空きがないことが検出されると、空き状況検出ステップ1bは、対象となるページ2bの前後のページ2a、2cの空き状況を、空き状況管理ステップ1aを参照することにより検出す

る。この例では、対象となるページ 2 b の前のページ 2 a には 20% の空きが、また、後のページ 2 c には 50% の空きが存在することが検出される。

【0023】次に、移動ステップ 1 c は、空き状況検出ステップ 1 b の検出結果と、挿入しようとするデータのデータ量とに応じて、データの移動先を決定する。いまの例では、挿入しようとするデータのデータ量が、ページ 2 b の 30% の容量に該当すると仮定すると、移動ステップ 1 c は、後方のページ 2 c を選択し、ページ 2 b の 30% の容量に該当するデータを、この後方のページ 2 c に対して移動させる。その結果、対象となるページ 2 b には、30% の空きが発生することになるので、この部分に対してデータを挿入することが可能となる。

【0024】以上に説明したように、本発明のデータ管理方法では、データの挿入対象となるページに空きがない場合には、その前後のページにデータを移動させるようにしたので、従来のように、新たなページを追加することなく、データを追加することが可能となるので、フラグメンテーションの発生を防止することができる。

【0025】次に、本発明の実施の形態の構成例について説明する。図 2 は、本発明の実施の形態の構成例を示す図である。この図に示すように、本実施の形態は、サーバ 30、データベース（以下、DB と称す）31～33、ネットワーク 34、クライアント 35～37 によって構成されている。

【0026】サーバ 30 は、CPU (Central Processing Unit) 30 a、ROM (Read Only Memory) 30 b、RAM (Random Access Memory) 30 c、HDD (Hard Disk Drive) 30 d、バス 30 e、I/F (Interface) 30 f、30 g によって構成されており、DB 31～33 を管理するとともに、クライアント 35～37 からアクセスがあった場合には、DB 31～33 に対してデータの読み書きを行う。

【0027】ここで、CPU 30 a は、HDD 30 d に格納されているプログラム等に従って、装置の各部を制御するとともに、各種演算処理を実行する。ROM 30 b は、CPU 30 a が実行する基本的なプログラムやデータを格納している。

【0028】RAM 30 c は、CPU 30 a が実行対象とするプログラムや、演算途中のデータを一時的に格納する。HDD 30 d は、CPU 30 a が実行するプログラムやデータを格納している。

【0029】バス 30 e は、CPU 30 a、ROM 30 b、RAM 30 c、HDD 30 d、および、I/F 30 f、30 g を相互に接続し、これらの間でデータの授受を可能とする。

【0030】I/F 30 f は、DB 31～32 とサーバ 30 とを電氣的に接続し、これらの間でデータを送受信する。I/F 30 g は、サーバ 30 とネットワーク 34 との間でデータを授受する際に、データ形式やプロトコ

ルの変換を行う。

【0031】DB 31～33 は、HDD 等の 2 次記憶装置によって構成されており、データと、データを管理するための索引としての B-Tree が格納されている。ここで、B-Tree を構成するページ（インデックスページおよびリーフページ）のそれぞれにはページ番号が付与されている。また、DB 31～33 のディレクトリには、図 3 (A) に示すような各ページの空き状況と、ページ番号とを関連付けた空き状況管理テーブルが格納されている。ここで、ページの空き状況は、2 ビットのデータによって表現され、図 3 (B) に示すように、空き状況が 0～19%（本実施の形態では空き状況を 1% 単位で表している）の場合が“11”、20～39% の場合が“10”、40～59% の場合が“01”、60%～100% の場合が“00”によってそれぞれ示される。なお、“00”の範囲が他に比べて広いのは、この範囲を広く設定することにより、後述するインデックスの移動処理や再統合処理等が頻繁に発生することを防止するためである。

【0032】ネットワーク 34 は、例えば、インターネットによって構成されており、クライアント 35～37 とサーバ 30 との間でデータを送受信する。クライアント 35～37 は、例えば、パーソナルコンピュータによって構成されており、サーバ 30 に対してアクセスし、目的のデータを読み書きする。

【0033】次に、以上の実施の形態の動作について説明する。まず、図 4 を参照して、リーフページ（以下、単に「ページ」と称す）に新たにインデックスを挿入する場合の動作について説明する。

【0034】いま、クライアント 35～37 の何れかから、新たなデータを DB 31 に追加する命令が送信されたとする。すると、サーバ 30 の CPU 30 a は、この命令を受信し、DB 31 の B-Tree を参照することにより、新たなデータに対応するインデックスを追加すべきページを特定する。

【0035】例えば、図 4 (A) に示すページ 51 が該当すると判定された場合には、CPU 30 a は、このページ 51 の空き状況を、空き状況管理テーブル（図 3 (A) 参照）を参照して、検出する。

【0036】図 5 (A) は、ページ 50～52 に対応する空き状況管理テーブルの一例である。いまの例では、ページ 51 には、空きが存在しないので、ページ指示値は“00”となっている。

【0037】該当するページ 51 に空きが存在しないことを認知した CPU 30 a は、その前後のページ 50、52 の空き状況を検出する。この例では、図 4 (A) に示すように後のページ 52 には 70% 程度の空きがあり、また、図 5 (A) に示すようにページ指示値が“10”となっていることから、CPU 30 a は、ページ 52 との間で、データの移動処理を実行し、必要な空き傾

域をページ51に確保する。

【0038】即ち、CPU30aは、インデックスを格納するために必要なデータ量を算出し、そのデータ量に見合うだけのインデックスを、ページ51からページ52に移動させる。いまの例では、必要な空き領域がページ51の40%に該当すると仮定すると、ページの全容量の40%分に該当するインデックスが、ページ52に移動される。

【0039】その結果、図4(B)に示すように、ページ51には、40%に相当する空き領域が確保されるので、CPU30aは、この領域に対して新たなデータに対応するインデックスを挿入する。そして、空き領域管理テーブルの該当する部分の指示値を、図5(B)に示すように書き換える。

【0040】以上の処理によれば、ページに対して新たなインデックスを追加する場合には、該当するページの空き領域を空き領域管理テーブルによって検出し、空き領域が不足している場合にはその前後のページに空き領域が存在しないか否かを判定し、空き領域が存在する場合には、データの一部をそのページに移動させた後、新たなインデックスを追加するようにした。その結果、前後のページに空き領域が存在する場合には、従来の方法のように新たなページを追加する必要がなくなるので、フラグメンテーションの発生を極力抑えることが可能となる。

【0041】また、データの挿入時にデフラグが自動的に実行されることになるので、従来のようにシステムを停止してデフラグを実行する必要がなくなる。その結果、管理者がシステムの状態を監視する手間を省略することも可能となる。

【0042】なお、以上の処理によりリーフページに格納されるインデックスが変化するので、それに応じてインデックスページに対してもインデックスを更新する必要がある。

【0043】次に、図6を参照して、ページからインデックスを削除する場合の動作について説明する。クライアント35~37の何れかから、所定のデータをDB31から削除する命令が送信されたとする。すると、サーバ30のCPU30aは、この命令を受信し、DB31のB-Treeを参照することにより、インデックスを削除すべきページを特定する。

【0044】例えば、図6(A)に示すページ51が該当すると判定された場合には、CPU30aは、該当するインデックスを削除するとともに、空き状況管理テーブルの該当する指示値を変更する。いまの例では、インデックスが削除された残りが図6(A)に示すように、50%の空きであるので、空き状況管理テーブルの指示値は“01”となる(図7(A)参照)。

【0045】次に、CPU30aは、削除の対象となったページ51の前後のページ50、52の空き状況をチ

ェックし、統合が可能か否かを判定する。いまの例では、後のページ52が70%の空き状態であるので、統合可能であるとし、CPU30aは、ページ51に格納されている全てのインデックスを、後のページ52に対して移動させる。その結果、図6(B)に示すように、ページ51は、100%空きの状態となるので、CPU30aは、これを再利用可能な「縮退ページ」として、システムに返還する。なお、このときの空き状況管理テーブルの一例を図7(B)に示す。

【0046】以上の処理によれば、ページからインデックスを削除する場合には、インデックスを削除した後のページと、その前後のページの空き領域を空き領域管理テーブルによって検出し、統合可能な場合にはインデックスを統合して空のページをシステムに返還するようにしたので、フラグメンテーションの発生を極力抑えることが可能となる。

【0047】なお、以上の処理により、前述の場合と同様に、リーフページに格納されるインデックスが変化するので、それに応じてインデックスページに対してもインデックスを更新する必要がある。

【0048】以上に説明したように、本発明の実施の形態によれば、インデックスの挿入または削除時に、対象ページの前後のページに対して、インデックスを適宜移動させるようにしたので、フラグメンテーションの発生を極力抑えることが可能となる。その結果、ページの個数の増大に起因する、必要な記憶領域の増大を防止するとともに、キー値として所定の範囲が与えられた場合においてページを横方向に順次検索する際のアクセス性能を向上させることが可能となる。

【0049】また、データの削除時にデフラグが自動的に実行されることになるので、従来のようにシステムを停止してデフラグを実行する必要がなくなる。その結果、管理者がシステムの状態を監視する手間を省略することも可能となる。

【0050】次に、図8および図9を参照して、以上の動作を実現するためのフローチャートについて説明する。図8は、新たなデータをデータベースに追加する際に、ページに対して実行される処理の一例である。このフローチャートが開始されると、以下の処理が実行される。

[S10] CPU30aは、データの追加の対象となるページを、B-Treeを参照して特定する。

【0051】[S11] CPU30aは、空き状況管理テーブルを参照して、ページの使用状況を取得する。

[S12] CPU30aは、取得した使用状況を参照し、空きがあるか否かを判定し、空きがある場合にはステップS18に進み、それ以外の場合にはステップS13に進む。

【0052】[S13] CPU30aは、対象となるページの前後に位置するページの使用状況を空き状況管理

テーブルから取得する。

【S14】CPU30aは、取得した使用状況を参照し、前後のページに空きがあるか否かを判定し、空きがある場合にはステップS17に進み、それ以外の場合にはステップS15に進む。

【0053】【S15】CPU30aは、対象ページの後に新規ページを挿入する。

【S16】CPU30aは、新規ページに対象ページのインデックスの一部を移動した後、対象ページにインデックスを挿入する。

【S17】CPU30aは、対象ページのインデックスを、前または後のページに移動させる。

【0054】なお、前後双方に空きがある場合には、コスト的に有利な後のページを選択する。

【S18】CPU30aは、対象ページにインデックスを挿入する。

【S19】CPU30aは、空き状況管理テーブルのページ使用状況を更新する。

【0055】次に、図9を参照して、データベースからデータを削除する場合に、ページに対して実行される処理の一例を説明するフローチャートである。このフローチャートが開始されると、以下の処理が実行される。

【S30】CPU30aは、データの削除の対象となるページを、B-Treeを参照して特定する。

【0056】【S31】CPU30aは、対象ページから該当するインデックスを削除する。

【S32】CPU30aは、空き状況管理テーブルの対象ページの使用状況を更新する。

【0057】【S33】CPU30aは、対象ページの前後のページの使用状況を空き状況管理テーブルから取得する。

【S34】CPU30aは、対象ページと、前または後のページの何れかを統合することが可能か否かを判定し、統合可能であると判定した場合にはステップS35に進み、それ以外の場合には処理を終了する。

【S35】CPU30aは、対象ページのインデックスを、統合可能と判定された前または後の何れかのページに対して移動させる。

【0058】なお、前または後の双方のページと統合可能であると判定された場合には、コスト的に有利な後のページを選択する。

【S36】CPU30aは、対象ページを再利用可能な縮退ページとしてシステムに返却する。

【S37】CPU30aは、空き状況管理テーブルの対象ページの使用状況を更新する。

【0059】以上の処理によれば、前述したように、フラグメンテーションの発生を抑制しながら、データの挿入または削除を行うことが可能となる。なお、以上の実施の形態では、B-Treeのページに対するインデックスの挿入または削除処理を例に挙げて説明したが、本

発明はこのような場合のみに限定されるものではなく、複数のデータから構成されるページが一定の論理関係に従って連続的に配置されてなる情報を管理する方法に対して適用することが可能である。

【0060】最後に、上記の処理機能は、コンピュータによって実現することができる。その場合、サーバが有すべき機能を、コンピュータで読み取り可能な記録媒体に記録されたプログラムに記述しておき、このプログラムをコンピュータで実行することにより実現することができる。コンピュータで読み取り可能な記録媒体としては、磁気記録装置や半導体メモリ等がある。市場へ流通させる場合には、CD-ROM(Compact Disk Read Only Memory)やフロッピー(登録商標)ディスク等の可搬型記録媒体にプログラムを格納して流通させたり、ネットワークを介して接続されたコンピュータの記憶装置に格納しておき、ネットワークを通じて他のコンピュータに転送することもできる。コンピュータで実行する際には、コンピュータ内のハードディスク装置等にプログラムを格納しておき、メインメモリにロードして実行する。

【0061】(付記1) 複数のデータから構成されるページが一定の論理関係に従って連続的に配置されてなる情報を管理する情報管理方法において、各ページの空き状況を管理する空き状況管理ステップと、何れかのページにデータを追加または削除する場合には、対象となるページの前後のページの空き状況を、前記空き状況管理ステップで得られた各ページの空き状況を参照して検出する空き状況検出ステップと、前記空き状況検出ステップによって検出された前後のページの空き状況に応じて、データの挿入前またはデータの削除後のページに含まれているデータを、前または後のページに移動させる移動ステップと、を有することを特徴とする情報管理方法。

【0062】(付記2) 前記移動ステップは、新たなデータを挿入する場合には、前記空き状況検出ステップによって検出された前後のページの空き状況に応じて、前記挿入対象となるページに属するデータを、前または後のページに移動させることを特徴とする付記1記載の情報管理方法。

【0063】(付記3) 前記移動ステップは、データを削除する場合には、前記空き状況検出ステップによって検出された前後のページの空き状況に応じて、削除の対象とならなかったデータを前または後のページに移動させることを特徴とする付記1記載の情報管理方法。

【0064】(付記4) 前記空き状況管理ステップは、空き領域が所定の容量以下である場合には、ページに空き領域が存在しないとみなすことを特徴とする付記1記載の情報管理方法。

【0065】(付記5) 前記移動ステップは、対象となるページの前後のページの双方に空きがある場合に

は、後のページに対してデータを移動させることを特徴とする付記1記載の情報管理方法。

【0066】(付記6) 前記ページの空き状況の管理は、空き状況に対応して分けられた複数の種別の何れかを示すデータをもとに行われることを特徴とする付記1記載の情報管理方法。

【0067】(付記7) 前記空き状況に対応して分けられた複数の種別のうち、最も空き状況が高い種別の範囲は、他の空き状況の種別の範囲に比べて大きいことを特徴とする付記6記載の情報管理方法。

【0068】(付記8) 複数のデータから構成されるページが一定の論理関係に従って連続的に配置されてなる情報をコンピュータに管理させるプログラムが記録されたコンピュータ読み取り可能な記録媒体において、コンピュータを、各ページの空き状況を管理する空き状況管理手段、何れかのページにデータを追加または削除する場合には、対象となるページの前後のページの空き状況を、前記空き状況管理手段を参照して検出する空き状況検出手段、前記空き状況検出手段によって検出された前後のページの空き状況に応じて、データの挿入前またはデータの削除後のページに含まれているデータを、前または後のページに移動させる移動手段、として機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体。

【0069】(付記9) 複数のデータから構成されるページが一定の論理関係に従って連続的に配置されてなる情報を管理する情報管理装置において、各ページの空き状況を管理する空き状況管理手段と、何れかのページにデータを追加または削除する場合には、対象となるページの前後のページの空き状況を、前記空き状況管理手段を参照して検出する空き状況検出手段と、前記空き状況検出手段によって検出された前後のページの空き状況に応じて、データの挿入前またはデータの削除後のページに含まれているデータを、前または後のページに移動させる移動手段と、を有することを特徴とする情報管理装置。

【0070】

【発明の効果】以上説明したように本発明では、複数のデータから構成されるページが一定の論理関係に従って連続的に配置されてなる情報を管理する情報管理方法において、各ページの空き状況を管理する空き状況管理ステップと、何れかのページにデータを追加または削除する場合には、対象となるページの前後のページの空き状況を、空き状況管理ステップで得られた各ページの空き状況を参照して検出する空き状況検出ステップと、空き状況検出ステップによって検出された前後のページの空き状況に応じて、データの挿入前またはデータの削除後のページに含まれているデータを、前または後のページに移動させる移動ステップと、を設けるようにしたので、データの挿入または削除が発生した場合に関連する

部分に対してデータの再配置処理を行うことが可能となり、24時間無停止で運用されるシステムにおいても、フラグメンテーションの発生を抑制することが可能となるとともに、アクセス性能を向上させることが可能となる。また、従来のように全ての領域に対してデフラグを行う場合に比較し、データの挿入または削除に応じて関連する部分のみについて再配置処理を行うことから、効率よくデフラグを行うことが可能となる。

【図面の簡単な説明】

10 【図1】本発明の動作原理を説明する原理図である。

【図2】本発明の実施の形態の構成例を示す図である。

【図3】(A)は空き状況管理テーブルの一例を示す図であり、(B)は空き状況管理テーブルの指示値を説明する図である。

【図4】ページにインデックスを挿入する場合の処理について説明する図であり、(A)が挿入前、(B)が挿入後の状態を示す図である。

【図5】図4に示すページに対応する空き状況管理テーブルの一例を示す図であり、(A)が挿入前、(B)が挿入後の状態を示す図である。

【図6】ページからインデックスを削除する場合の処理について説明する図であり、(A)が削除前、(B)が削除後の状態を示す図である。

【図7】図6に示すページに対応する空き状況管理テーブルの一例を示す図であり、(A)が削除前、(B)が削除後の状態を示す図である。

【図8】ページにインデックスを挿入する場合に実行される処理の一例である。

30 【図9】ページからインデックスを削除する場合に実行される処理の一例である。

【図10】B-Tre eの構造を示す図である。

【図11】従来における、ページにインデックスを挿入する場合の処理について説明する図である。

【図12】従来における、ページからインデックスを削除する場合の処理について説明する図である。

【符号の説明】

1 サーバ

1 a 空き状況管理ステップ

1 b 空き状況検出ステップ

40 1 c 移動ステップ

2 データベース

2 a ~ 2 c ページ

30 サーバ

30 a CPU

30 b ROM

30 c RAM

30 d HDD

30 e バス

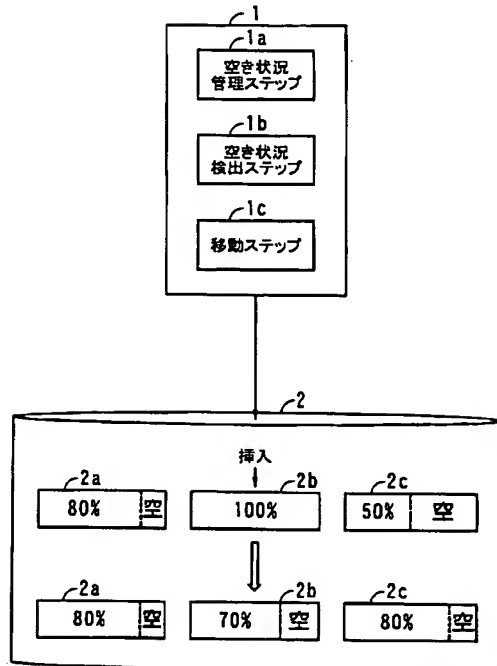
30 f, 30 g I/F

50 31 ~ 33 DB

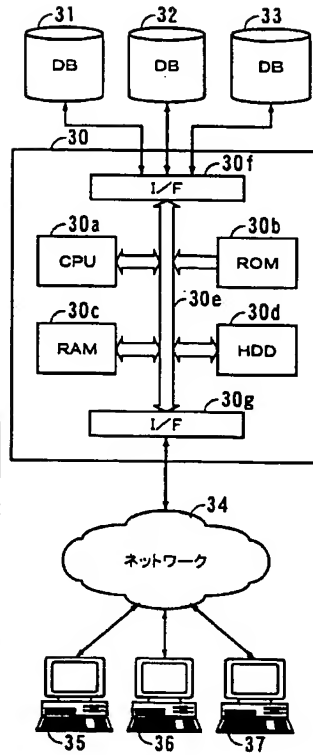
3.4 ネットワーク

3.5～3.7 クライアント

【図1】



【図2】



【図3】

ページ番号	指示値
0001	11
0002	00
0003	10
...	...
n	01

(A)

指示値	使用率
00	60%～100%
01	40%～59%
10	20%～39%
11	0%～19%

(B)

【図7】

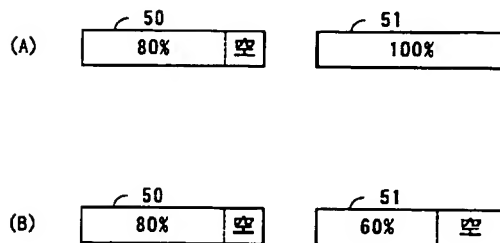
ページ番号	ページ指示値
50	00
51	01
52	10

(A)

ページ番号	ページ指示値
50	00
51	—
52	00

(B)

【図4】



【図5】

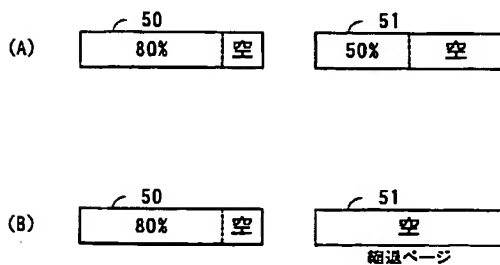
ページ番号	ページ指示値
50	00
51	00
52	10

(A)

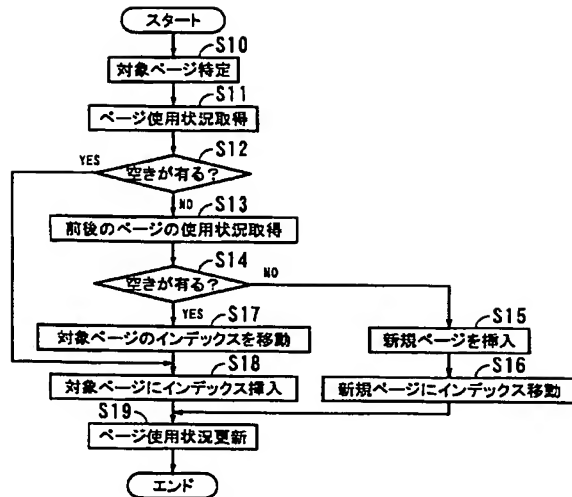
ページ番号	ページ指示値
50	00
51	00
52	00

(B)

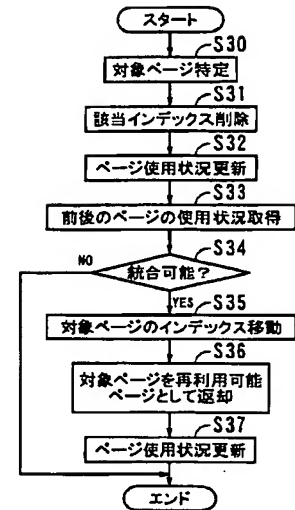
【図6】



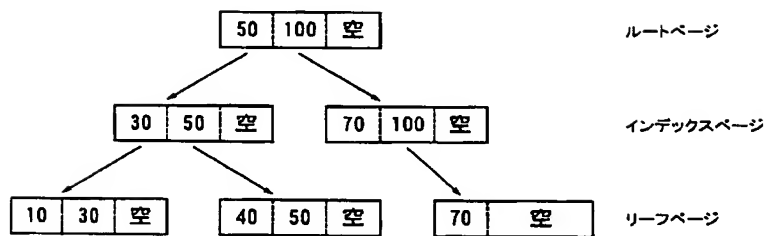
【図 8】



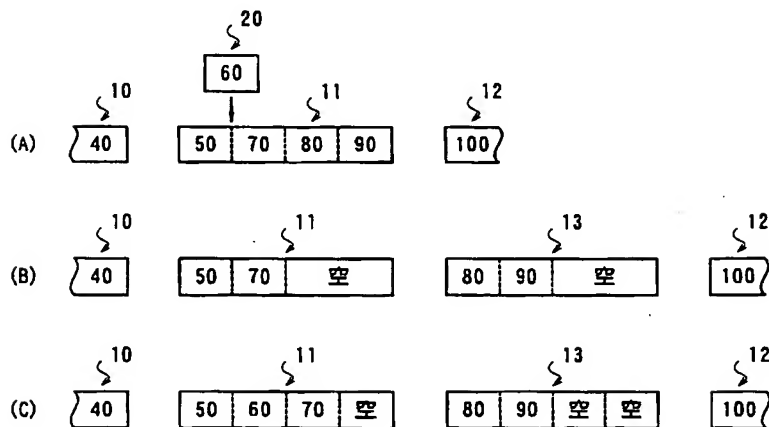
【図 9】



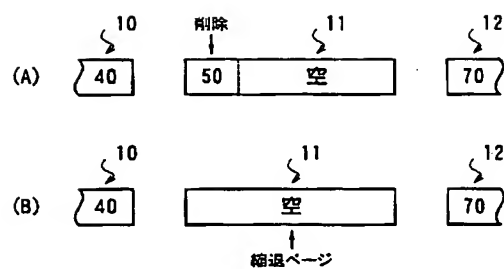
【図 10】



【図 11】



【図 12】



フロントページの続き

(72)発明者 関谷 修志

神奈川県小田原市栄町一丁目 6 番地 4 株
 式会社富士通ソフトウェア生産技術研究所
 内

Fターム(参考) 5B065 BA01 CH18 ZA15

5B075 NR03

5B082 BA11 CA02 CA08 EA01

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.